Published in: Levin, Lori/Stede, Manfred (Eds.): Proceedings of the 8th Linguistic Annotation Workshop in conjunction with COLING 2014 (LAW-VIII). August 23-24, 2014. Dublin, Ireland. - Stroudsburg, PA: ACL, 2014. Pp. 20-28.

POS error detection in automatically annotated corpora

Ines Rehbein
SFB 632 Information Structure
German Department
Potsdam University
irehbein@uni-potsdam.de

Abstract

Recent work on error detection has shown that the quality of manually annotated corpora can be substantially improved by applying consistency checks to the data and automatically identifying incorrectly labelled instances. These methods, however, can not be used for automatically annotated corpora where errors are systematic and cannot easily be identified by looking at the variance in the data. This paper targets the detection of POS errors in automatically annotated corpora, so-called *silver standards*, showing that by combining different measures sensitive to annotation quality we can identify a large part of the errors and obtain a substantial increase in accuracy.

1 Introduction

Today, linguistically annotated corpora are an indispensable resource for many areas of linguistic research. However, since the emergence of the first digitised corpora in the 60s, the field has changed considerably. What was considered "very large" in the last decades is now considered to be rather small. Through the emergence of Web 2.0 and the spread of user-generated content, more and more data is accessible for building corpora for specific purposes.

This presents us with new challenges for automatic preprocessing and annotation. While conventional corpora mostly include written text which complies to grammatical standards, the new generation of corpora contain texts from very different varieties, displaying features of spoken language, regional variety, ungrammatical content, typos and non-canonical spelling. A large portion of the vocabulary are unknown words (that is, not included in the training data). As a result, the accuracy of state-of-the-art NLP tools on this type of data is often rather low. In combination with the increasing corpus sizes, it seems that we have to lower our expectations with respect to the quality of the annotations. Time-consuming double annotation or a manual correction of the whole corpus is often not feasible. Thus, the use of so-called *silver standards* has been discussed (Hahn et al., 2010; Kang et al., 2012; Paulheim, 2013), along with their adequacy to replace carefully hand-crafted gold standard corpora.

Other approaches to address this problem come from the areas of domain adaptation and error detection. In the first field, the focus is on adapting NLP tools or algorithms to data from new domains, thus increasing the accuracy of the tools. In error detection, the goal is to automatically identify erroneous labels in the data and either hand those instances to a human annotator for manual correction, or to automatically correct those cases. Here, the focus is not on improving the tools but on increasing the quality of the corpus and, at the same time, reducing human effort. These approaches are not mutually exclusive but can be seen as complementary methods for building high-quality language resources at a reasonable expense.

We position our work at the interface of these fields. Our general objective is to build a high-quality linguistic resource for informal spoken youth language, annotated with parts of speech (POS) information. As we do not have the resources for proofing the whole corpus, we aim at building a silver standard where the quality of the annotations is high enough to be useful for linguistic research. For automatic

preprocessing, we use tagging models adapted to our data. The main contribution of this paper is in developing and evaluating methods for POS error detection in automatically annotated corpora. We show that our approach not only works for our data but can also be applied to canonical text from the newspaper domain, where the POS accuracy of standard NLP tools is quite high.

The paper is structured as follows. Section 2 reviews related work on detecting annotation errors in corpora. Section 3 describes the underlying assumptions of our approach. In Section 4, we describe the experimental setup and data used in our experiments, and we present our results in Section 5. We conclude in Section 6.

2 Related Work

Most work on (semi-)automatic POS error detection has focussed on identifying errors in POS assigned by *human annotators* where variation in word-POS assignments in the corpus can be caused either by ambiguous word forms which, depending on the context, can belong to different word classes, or by incorrect judgments made by the annotators (Eskin, 2000; van Halteren, 2000; Květoň and Oliva, 2002; Dickinson and Meurers, 2003; Loftsson, 2009).

The *variation n-gram algorithm* (Dickinson and Meurers, 2003) allows users to identify potentially incorrect tagger predictions by looking at the variation in the assignment of POS tags to a particular word ngram. The algorithm produces a ranked list of varying tagger decisions which have to be processed by a human annotator. Potential tagger errors are positioned at the top of the list. Later work (Dickinson, 2006) extends this approach and explores the feasibility of automatically correcting these errors.

Eskin (2000) describes a method for error identification using *anomaly detection*, where anomalies are defined as elements coming from a distribution different from the one in the data at hand. Květoň and Oliva (2002) present an approach to error detection based on a semi-automatically compiled list of *impossible ngrams*. Instances of these ngrams in the data are assumed to be tagging errors and are selected for manual correction.

All these approaches are tailored towards identifying human annotation errors and cannot be applied to our setting where we have to detect systematic errors made by automatic POS taggers. Thus, we can not rely on *anomalies* or *impossible ngrams* in the data, as the errors made by the taggers are consistent and, furthermore, our corpus of non-canonical spoken language includes many structures which are considered *impossible* in Standard German.

Rocio et al. (2007) address the problem of finding systematic errors in POS tagger predictions. Their method is based on a modified multiword unit extraction algorithm which extracts cohesive sequences of tags from the corpus. These sequences are then sorted manually into linguistically sound ngrams and potential errors. This approach addresses the correction of large, automatically annotated corpora. It successfully identifies (a small number of) incorrectly tagged high-frequency sequences in the text which are often based on tokenisation errors. The more diverse errors due to lexical ambiguity, which we have to deal with in our data, are not captured by this approach.

Most promising is the approach of Loftsson (2009) who evaluates different methods for error detection, including an ensemble of five POS taggers, where error candidates are defined as those instances for which the predictions of the five taggers disagree. His method successfully identifies POS errors and thus increases the POS accuracy in the corpus. Using the tagger ensemble, Loftsson (2009) is able to identify error candidates with a precision of around 16%. He does not report recall, that is how many of the erroneously tagged instances in the corpus have been found. We apply the ensemble method to our data and use it as our baseline.

Relevant to us is also the work by Dligach and Palmer (2011), who show how the need for double annotation can be efficiently reduced by only presenting carefully selected instances to the annotators for a second vote. They compare two different selection methods. In the first approach, they select all instances where a machine learning classifier disagrees with the human judgement. In the second approach, they use the probability score of a maximum entropy classifier, selecting instances with the smallest *prediction margin* (the difference between the probabilities for the two most probable predictions). Dligach and Palmer (2011) test their approach in a Word Sense Disambiguation task. The main

ideas of this work, however, can be easily applied to POS tagging.

3 Identifying Systematic POS Errors

Taggers make POS errors for a number of reasons. First of all, anomalies in the input can cause the tagger to assign an incorrect tag, e.g. for noisy input with spelling or tokenisation errors. Another source of errors are out-of-vocabulary words, that is word forms unknown to the tagger because they do not exist in the training data. A third reason for incorrect tagger judgments are word forms which are ambiguous between different parts of speech. Those cases can be further divided into cases where the information for identifying the correct label is there but the tagger does not make use of it, and into cases that are truly ambiguous, meaning that even a human annotator would not be able to disambiguate the correct POS tag. Tagger errors can also be caused by ill-defined annotation schemes or errors in the gold standard (see Manning (2011) for a detailed discussion on different types of POS errors).

To assess the difficulty of the task, it might be interesting to look at the agreement achieved by human annotators for POS tagging German. The inter-annotator agreement for POS annotation with the STTS on written text is quite high with around 0.97-0.98 Fleiss κ, and for annotating spoken text using an extended version of the STTS similar numbers can be obtainded (Rehbein and Schalowski, 2013).

In this work, we are not so much interested in finding tokenisation and spelling errors but in identifying automatic tagger errors due to lexical ambiguity. Our work is based on the following assumptions:

> **Assumption 1:** Instances of word forms which are labelled differently by different taggers are potential POS errors.

> **Assumption 2:** POS tags which have been assigned with a low probability by the tagger are potential POS errors.

In the remainder of the paper, we present the development of a system for error detection and its evaluation on a corpus of informal, spontaneous dialogues and on German newspaper text. We report precision and recall for our system. *Precision* is computed as the number of correctly identified error candidates, divided by the number of all (correctly and incorrectly identified) error candidates *(number of true positives / (number of true positives + false positives))*, and *recall* by dividing the number of identified errors by the total number of errors in the data *(true positives / (true positives + false negatives))*.

4 Experimental Setup

The data we use in our experiments comes from two sources, i) from a corpus of informal, spoken German youth language (The KiezDeutsch Korpus (KiDKo) Release 1.0) (Rehbein et al., 2014), and ii) from the TIGER corpus (Brants et al., 2002), a German newspaper corpus.

4.1 Kiezdeutsch – Informal youth language

KiDKo is a new language resource including informal, spontaneous dialogues from peer-to-peer communication of adolescents. The current version of the corpus includes the audio signals aligned with transcriptions, as well as a normalisation layer and POS annotations. Additional annotation layers (Chunking, Topological Fields) are in progress.

The transcription scheme has an orthographic basis but, in order to enable investigations of prosodic characteristics of the data, it also tries to closely capture the pronunciation, including pauses, and encodes disfluencies and primary accents. On the normalisation layer, non-canonical pronunciations and capitalisation are reduced to standard German spelling. The normalisation is done on the token level, and non-canonical word order as well as disfluencies are included in the normalised version of the data (Example 1).

(1) [transcription]: isch hab au (–) isch hab isch hab auch äh FLATrate
[normalisation]: Ich habe au # PAUSE Ich habe , ich habe auch äh Flatrate .
I have too # I have , I have too uh flatrate .
"I have ... I have, I have a flatrate, too." (MuH23MT)

	KiDKo			TIGER	
Baseline taggers	avg. (5-fold)	dev	test	dev	test
Brill	94.4	94.7	93.8	96.8	96.8
Treetagger	95.1	95.5	94.8	97.2	97.4
Stanford	95.3	95.6	94.7	97.4	97.5
Hunpos	95.6	95.8	94.8	97.4	97.5
CRF	**96.9**	**97.4**	**96.1**	**97.9**	**98.0**

Table 1: Baseline results for different taggers on KiDKo and TIGER (results on KiDKo are given for a 5-fold cross validation (5-fold) and for the development and test set)

We plan to release the POS tagged version of the corpus in summer 2014. Due to legal constraints, the audio files will have restricted access and can only be accessed locally while the transcribed and annotated version of the corpus will be available over the internet via ANNIS (Zeldes et al., 2009).[1]

4.2 The TIGER corpus

The second corpus we use in our experiments is the TIGER corpus (release 2.2), a German newspaper corpus with approximately 50,000 sentences (900,000 tokens). We chose TIGER to show that our approach is not tailored towards one particular text type but can be applied to corpora of different sizes and from different domains.

4.3 Baseline

In our experiments, we use a subpart of KiDKo with 103,026 tokens, split into a training set with 66,024 tokens, a development set with 16,530 tokens, and a test set with 20,472 tokens. The TIGER data was also split into a training set (709,740 tokens), a development set (88,437 tokens) and a test set (90,061 tokens).

To test our first assumption, we trained an ensemble of five taggers on the two corpora (see list below), and checked all instances where the taggers disagreed. We consider all cases as disagreements where at least one of the five taggers made a prediction different from the other taggers.

The five taggers we use reflect different approaches to POS tagging (including Transformation-based Learning, Markov Models, Maximum Entropy, Decision Trees, and Conditional Random Fields):

- the Brill tagger (Brill, 1992)
- the Hunpos tagger[2]
- the Stanford POS tagger (Toutanova and Manning, 2000)
- the Treetagger (Schmid, 1995)
- a CRF-based tagger, using the CRFSuite[3]

Table 1 shows the accuracies of the different taggers on KiDKo and on TIGER (because of the smaller size of KiDKo, we also report numbers from a 5-fold cross validation on the training data). The CRF-based tagger gives the best results on the spoken language data as well as on TIGER. For more details on the implementation and features of the CRF tagger, please refer to (Rehbein et al., 2014).

For the KiDKo development set, we have 1,228 cases where the taggers disagree, that is 1,228 error candidates, and 1,797 instances in the test set. Out of those, 267 (dev) and 558 (test) are true errors (Table 2). This means that the precision of this simple heuristic is between 21.7% and 33%, with a recall between 61.1 and 70.8%. For TIGER, precision and recall are higher. Applying this simple heuristic, we are able to identify around 70% of the errors in the data, with a precision of around 27%. We consider this as our baseline.

[1] ANNIS (ANNotation of Information Structure) is a corpus search and visualisation interface which allows the user to formulate complex search queries which can combine multiple layers of annotation.

[2] The Hunpos tagger is an open source reimplementation of the TnT tagger (`https://code.google.com/p/hunpos`)

[3] `http://www.chokkan.org/software/crfsuite/`

	tokens	candidates	true err.	out of	% prec	% rec.
			KiDKo			
dev	16,530	1,228	267	437	21.7	61.1
test	20,472	1,797	558	788	33.0	70.8
			TIGER			
dev	88,437	4,580	1,280	1,818	27.9	70.4
test	90,061	4,618	1,246	1,754	27.0	71.0

Table 2: Number of error candidates identified by the disagreements in the ensemble tagger predictions (baseline)

5 Finding measures for error detection

When defining measures for error detection, we have to balance precision against recall. Depending on our research goal and resources available for corpus creation, we might either want to obtain a high precision, meaning that we only have to look at a small number of instances which are most probably true POS errors, or we might want to build a high-quality corpus where nearly all errors have been found and corrected, at the cost of having to look at many instances which are mostly correct.

5.1 Increasing precision

First, we try to improve precision and thus to reduce the number of false positives we have to look at during the manual correction phase. We do this by training a CRF classifier to detect errors in the output of the ensemble taggers. The features we use are shown in Table 3 and include the word form, the tags predicted by the tagger ensemble, ngram combinations of the ensemble POS tags, word and POS context for different context windows for the POS predicted by the CRF tagger and the Treetagger, a combination of word form and POS context (for CRF, Treetagger, and combinations of both; for window sizes of 3 and 4 with varying start and end positions), and the class label (1: error, 0: correct).

We experimented with different feature combinations and settings. Our basic feature set gives us high precision on both data sets, with very low recall. Only around 4-6% of all errors are found. However, precision is between 55-65%, meaning that the majority of the selected candidates are true errors.

Our extended feature sets (I and II) aim at improving recall by alleviating the sparse data problem. The extended feature set I extracts new features where the tags from the fine-grained German tagset, the STTS (Schiller et al., 1999), are converted into the coarse-grained universal tagset of Petrov et al. (2012),

basic features	**example**
word form	der (the)
lowercased word form	der
ensemble tags	PDS ART PDS PDS ART
POS context (CRF)	ADV:PROAV:VAFIN:APPR, PROAV:VAFIN:APPR:PDS, ...
POS context (tree)	PROAV:VAFIN:APPR, VAFIN:APPR:ART, ...
word form with POS context (CRF)	PROAV:VAFIN:APPR:der, VAFIN:APPR:der:APPR, ...
word form with POS context (CRF:tree)	PROAV:VAFIN:APPR:der, ..., der:APPR:ART:NN, ...
extended features I: universal POS	
universal ensemble tags	P D P P D
universal POS ngrams	P:D, P:P, P:P, ..., P:P:P:D, P:D:P:P:D
universal POS context (CRF)	ADV:P:VF:ADP, P:VF:ADP:P, ...
word form with universal POS context (CRF)	P:VF:ADP:der, VF:ADP:der:ADP, ADP:der:ADP:D, ...
word form with universal POS context (CRF:tree)	VF:VF:ADP:ADP:der, ADP:ADP:der:ADP:ADP, ...
extended features II: brown clusters	
brown cluster for word form	110111011111
brown cluster with universal POS context (CRF)	ADV:P:110111111110:ADP, P:110111111110:ADP:P, ...
class label (1 or 0)	1

Table 3: Features used for error detection

	tokens	candidates	true err.	out of	% prec	% rec
KiDKo						
basic features						
dev	16,530	32	21	437	65.6	4.8
test	20,472	59	32	788	54.2	4.1
extended features I (universal POS)						
dev	16,530	77	38	437	49.3	8.7
test	20,472	172	88	788	51.2	11.2
extended features II (universal POS, Brown clusters)						
dev	16,530	88	50	437	56.8	11.4
test	20,472	205	104	788	50.7	13.2
TIGER						
basic features						
dev	88,437	163	101	1,818	62.0	5.6
test	90,061	202	111	1,754	54.9	6.3
extended features I (universal POS)						
dev	88,437	564	348	1,818	61.7	19.1
test	90,061	588	347	1,754	59.0	19.8
extended features II (universal POS, Brown clusters)						
dev	88,437	501	318	1,818	63.5	17.5
test	90,061	518	298	1,754	57.5	17.0

Table 4: Number of error candidates identified by the classifier, precision (prec) and recall (rec)

with minor modifications.[4] On KiDKo, the universal POS features increase recall from around 5% up to 8-14%. On TIGER, the results are more substantial. Here, our recall increases from 5-6% up to nearly 20%, while precision is still in the same range (Table 4).

Our basic features were designed to add more (local) context useful for disambiguating between the different tags. Especially the right context (assigned POS) includes information which often helps, e.g. when distinguishing between a substitutive demonstrative pronoun (PDS) and a determiner (ART), which is a frequent error especially in the spoken language data.

We try to achieve further improvements by adding new features where we replace the word forms with Brown word cluster paths (Brown et al., 1992).[5] The extended features are designed to address the unknown word problem by generalising overvi t word forms. On the smaller KiDKo data set, this again has a positive effect, increasing both precision and recall. On TIGER, however, the results are mixed, with a higher precision on the development set but a somewhat lower recall for both, development and test sets. This is not surprising, as semi-supervised techniques are expected to help most for settings where data sparseness is an issue.

Overall, our error detection classifier is able to identify errors in the corpus with a good precision, meaning that only a small number of instances have to be checked manually in order to achieve an error rate reduction in the range of 11-17%. This approach seems suitable when limited resources are available for manual correction, thus asking for a method with high precision and low time requirements.

5.2 Increasing recall

While our attempts to increase precision were quite successful, we had to put up with a severe loss in recall. However, we would like to keep precision reasonably high but also to increase recall. Our next approach takes into account the marginal probabilities of the predictions (0: correct/1: error) of the CRF-based error detection classifier. We not only check those instances which the classifier has labelled as

[4]For instance, instead of converting all verb tags to V, we keep a tag for finite verbs (VF).

[5]The word clusters have been trained on the Huge German Corpus (HGC) (Fitschen, 2004), using a cluster size of 1000, a frequency threshold of 40 and a maximum path length of 12.

	tokens	threshold	candidates	true err.	out of	% prec	% rec
			KiDKo				
			extended features II (universal POS, Brown clusters)				
dev	16,530	0.8	286	120	437	42.0	27.5
dev	16,530	0.85	350	138	437	39.4	31.6
test	20,472	0.8	472	190	788	40.2	24.1
test	20,472	0.85	561	227	788	40.5	28.8
			TIGER				
			extended features I (universal POS)				
dev	88,437	0.8	1,208	602	1,818	49.8	33.1
dev	88,437	0.85	1,431	658	1,818	46.0	36.2
test	90,061	0.8	1,276	605	1,754	47.4	34.5
test	90,061	0.85	1,554	670	1,754	43.1	38.2

Table 5: Number of error candidates identified by the classifier using a marginal probability threshold

incorrect, but also those which have been labelled as correct, but with a marginal probability below a particular threshold. Table 5 gives results for a threshold of 0.8 and 0.85, using the best-scoring feature sets from the last experiment.

Our new measure results in a substantial increase in recall. Setting the threshold to 0.85, we are now able to detect around 30% of the errors in KiDKo and 36 to 38% in TIGER, while precision is still reasonably high. Figure 1 shows the relation between precision and recall for different thresholds from 0.95 to 0.1. Setting the threshold to 0.8, for example, would result in an error prediction presicion of around 40-42% for KiDKo and of around 47-50% for TIGER. Recall for error identification using a threshold of 0.8 would be in the range of 24-27.5% for KiDKo and 33-34.5% for TIGER. If we wanted to increase recall up to 50% for KiDKo, we would have to use a marginal probability threshold of approximately 0.65, and precision would drop to around 14%. This knowledge allows us to make an informed decision during corpus compilation, either starting from the POS accuracy we want to achieve, or from the resources we have for manual correction, and to predict the POS accuracy of the final corpus.

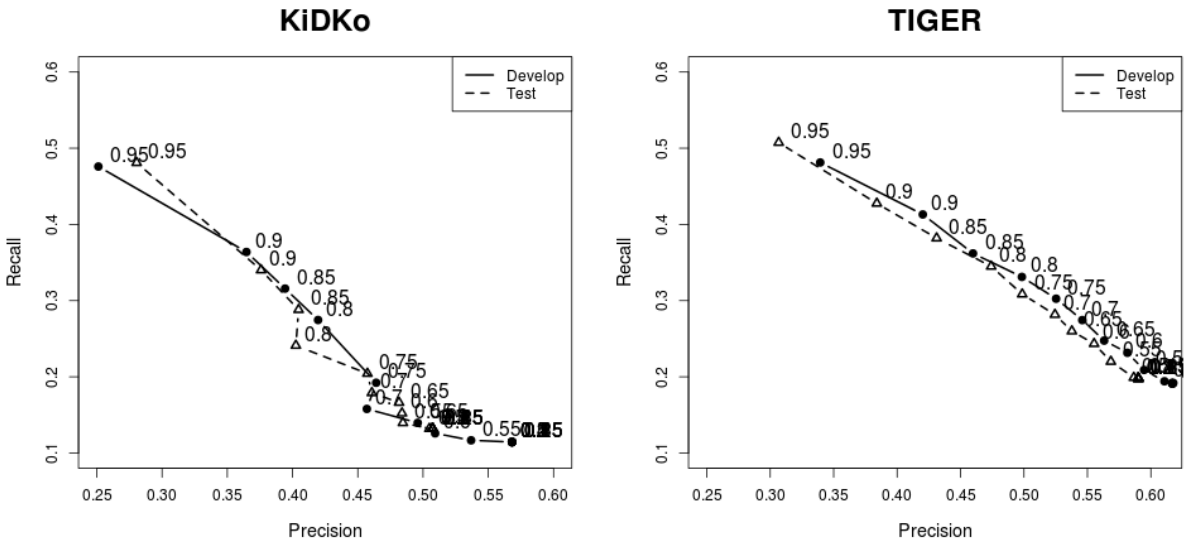

Figure 1: Trade-off between precision and recall for different marginal probability thresholds

6 Conclusions

In the paper, we presented and evaluated a system for automatic error detection in POS tagged corpora, with the goal of increasing the quality of so-called *silver standards* with minimal human effort. Our baseline, a simple heuristic based on disagreements in tagger predictions, allows us to identify between 60 and 70% of all errors in our two data sets, but with a low precision. We show how to refine this method, training a CRF-based classifier which is able to identify POS errors in tagger output with a much higher precision, thus reducing the need for manual correction.

Our method is able to find different types of POS errors, including the ones most frequently made by the tagger (adjectives, adverbs, proper names, foreign language material, finite verbs, verb particles, and more). Furthermore, it allows us to define the parameters which are most adequate for the task at hand, either aiming at high precision at the cost of recall, or increasing recall (and thus the annotation quality of the corpus) at the cost of greater manual work load. In addition, our method can easily be applied to different corpora and new languages.

References

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the First Workshop on Treebanks and Linguistic Theories*, pages 24–42.

Eric Brill. 1992. A simple rule-based part of speech tagger. In *3rd conference on Applied natural language processing (ANLC'92)*, Trento, Italy.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Markus Dickinson and Detmar W. Meurers. 2003. Detecting errors in part-of-speech annotation. In *10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*.

Markus Dickinson. 2006. From detecting errors to automatically correcting them. In *Annual Meeting of The European Chapter of The Association of Computational Linguistics (EACL-06)*, Trento, Italy.

Dmitriy Dligach and Martha Palmer. 2011. Reducing the need for double annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, LAW V '11.

Eleazar Eskin. 2000. Automatic corpus correction with anomaly detection. In *1st Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Seattle, Washington.

Arne Fitschen. 2004. *Ein computerlinguistisches Lexikon als komplexes System*. Ph.D. thesis, Institut für Maschinelle Sprachverarbeitung der Universität Stuttgart.

U. Hahn, K. Tomanek, E. Beisswanger, and E. Faessler. 2010. A proposal for a configurable silver standard. In *The Fourth Linguistic Annotation Workshop*, LAW 2010, pages 235–242.

Ning Kang, Erik van Mulligen, and Jan Kors. 2012. Training text chunkers on a silver standard corpus: can silver replace gold? *BMC Bioinformatics*, 13(1):17.

Pavel Květoň and Karel Oliva. 2002. (Semi-)Automatic detection of errors in PoS-tagged corpora. In *19th International Conference on Computational Linguistics (COLING-02)*.

Hrafn Loftsson. 2009. Correcting a POS-tagged corpus using three complementary methods. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, Athens, Greece, March.

Christopher D. Manning. 2011. Part-of-speech tagging from 97linguistics? In *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing - Volume Part I*, CICLing'11, pages 171–189.

Heiko Paulheim. 2013. Dbpedianyd - a silver standard benchmark dataset for semantic relatedness in dbpedia. In *CEUR Workshop*, CEUR Workshop Proceedings. CEUR-WS.org.

Slav Petrov, Dipanjan Das, and Ryan T. McDonald. 2012. A universal part-of-speech tagset. In *The Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2089–2096.

Ines Rehbein and Sören Schalowski. 2013. STTS goes Kiez – Experiments on annotating and tagging urban youth language. *Journal for Language Technology and Computational Linguistics*.

Ines Rehbein, Sören Schalowski, and Heike Wiese. 2014. The KiezDeutsch Korpus (KiDKo) release 1.0. In *The 9th International Conference on Language Resources and Evaluation (LREC-14)*, Reykjavik, Iceland.

Vitor Rocio, Joaquim Silva, and Gabriel Lopes. 2007. Detection of strange and wrong automatic part-of-speech tagging. In *Proceedings of the Aritficial Intelligence 13th Portuguese Conference on Progress in Artificial Intelligence*, EPIA'07.

Anne Schiller, Simone Teufel, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textkorpora mit STTS. Technical report, Universität Stuttgart, Universität Tübingen.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to German. In *ACL SIGDAT-Workshop*.

Kristina Toutanova and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the conference on Empirical methods in natural language processing and very large corpora*, EMNLP '00, Hong Kong.

Hans van Halteren. 2000. The detection of inconsistency in manually tagged text. In *Proceedings of the COLING-2000 Workshop on Linguistically Interpreted Corpora*, Centre Universitaire, Luxembourg, August.

Amir Zeldes, Julia Ritz, Anke Lüdeling, and Christian Chiarcos. 2009. Annis: A search tool for multi-layer annotated corpora. In *Corpus Linguistics 2009*.